NAME: ARNAB KUMAR MONDAL

UIN: 326006709.

ECEN 654. LAB 2:

Simulated Annealing Algorithm:

Algorithm 10.1 Simulated Annealing Algorithm for Floorplanning

```
1. Get an initial floorplan S;   S_Best = S;
2. Get an initial temperature T > 0;
3. while not "frozen" do
4.     for i = 1 to k do
5.       Perturb the floorplan to get a neighboring S' from S;
6.       ΔC = cost(S') – cost(S);
7.       if ΔC ≤ 0 then    // down-hill move
8          S = S';
9.       else    // uphill move
10.        S = S' with the probability e^(-ΔC/T);
11.      end if
12.      if cost(S_Best) > cost(S) then
13.       (S_Best) = S;
14.      end if
15.   end for
16.   T = rT;    // reduce temperature
17. end while
18. return S_Best;
```

After generating the simulated annealing algorithm the polish expression evaluation and traversal order are as follows

Algorithm 10.3 Polish Expression Evaluation (E)

```
1. stack s;
2. for i = 1 to 2n-1 do
3.       if e_i is an operand then s.push(e_i);
4.       if e_i is an operator then
5.          a = s.pop( ); b = s.pop( ); c = a e_i b;
6.          s.push(c);
7.       end if
8. end for
9. return s.pop( );
```

Algorithm 10.2 PostOrderTraversal(T)

```
1. /* Post-order traversal of a binary tree */
2. if root(T) ! = NULL then
3.    PostOrderTraversal(LeftSubtree(T));
4.    PostOrderTraversal(RightSubtree(T));
5.    Visit(root(T));
6. end if
7. return;
```

The Move Set:

- Floorplan solutions are represented by normalized Polish expressions.
- Three types of moves are suggested to perturb a given normalized Polish expression:

M_1: swap two adjacent operands;

M_2: complement some chain of nonzero length; (where $\overline{V} = H$ and $\overline{H} = V$);

M_3: swap two adjacent operand and operator.

PROGRAM METHODOLOGY:

In the program we used the main function where we call the move operation functions as well as the cost function. The data is taken from the excel files and it contains information regarding each individual modules, the pio's and the netlist associated with the given floorplanning. The main function does the simulated annealing algorithm and displays the floorplanning and the initial and the final cost of the given floorplanning. Initially a random cutset is derived by using 2 numeric values [-5,-6] as the vertical and the horizontal cut following which we evaluate our expression and cut cost. The probability of crossing the function is determined by the temperature variable T which changes steadily and thus changes the probability.

The normalized polish expression hold that if the number of operators are n then the number of elements in the polish expression is (2n+1). Since there are 100 modules we have 149 elements in our polish expression.

Main_cost function : This takes the primary input output data and helps us evaluate the cost function using the following formula

Cost Function = area + 10* aspectRatio + 3 * wirelen + 6 * iolen

Where area is the area of the given bounding box, wirelen is the estimated wirelength calculation using the half perimeter formula and iolen is an estimate of the sum of rectilinear routing distances to the nearest edge of the smallest rectangle which encloses the layout (shortest distance to possible io pin for each module listed in pio file). The aspect ratio can be defined as : asp(i)= abs(H(i)-W(i)) / max([H(i) W(i)]); where H(i) and W(i) are defined as the height and width of the individual modules.

M_1 function : This helps us to take an old polish expression and change it into a new polish expression by swapping the adjacent operands and sending it across to the main function to find the cost of the new polish expression.

M_2 function : This helps us to take an old polish expression and change it into a new polish expression by complementing the polish operators in a non zero chain length by changing our negative numeric values which stand for V and H in the normalized polish expression. sending it across to the main function to find the cost of the new polish expression.

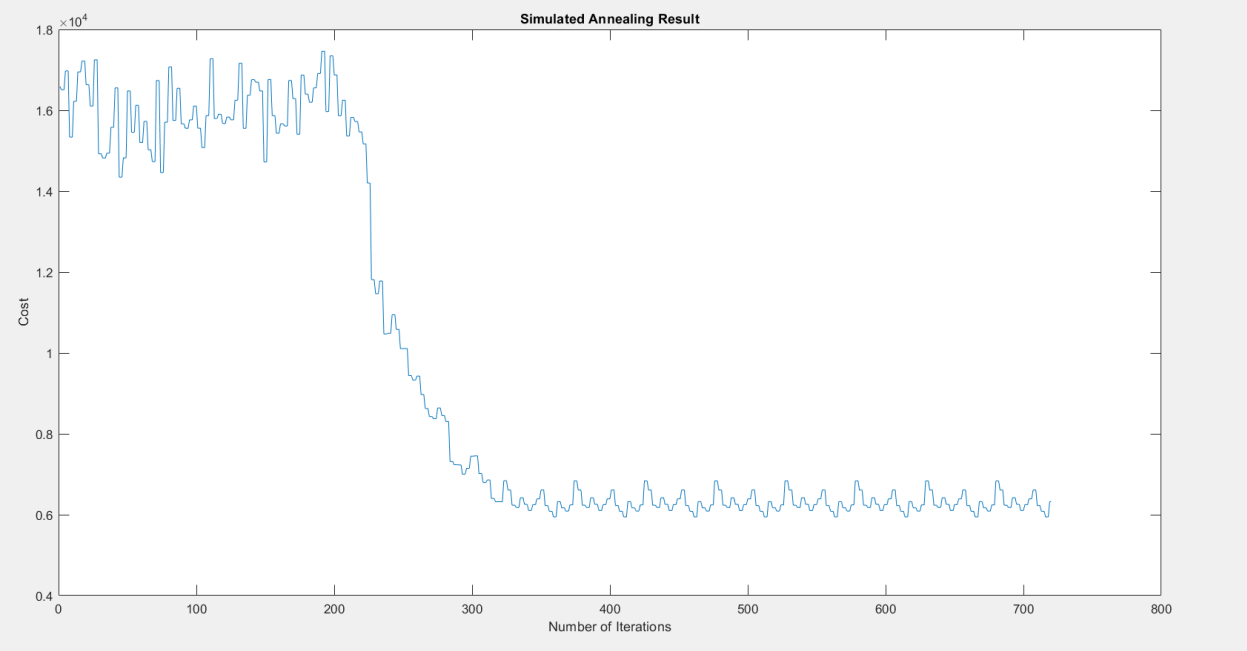
M_3 function : This helps us to take an old polish expression and change it into a new polish expression by swapping adjacent operators and operands and sending it across to the main function to find the cost of the new polish expression.

Results : For the simulated annealing problem the intital and final costs are as follows

```
initial cost
   1.0387e+04

final cost
   6.3235e+03
```

The cost curve with respect to the iteration graphs is :



Conclusion and further improvements : We can see that if we change the temperature parameters or the cost function we might be able to optimize it further but it should be done till a reasonable limit as the simulated annealing algorithm is a heuristic based algorithm and thus is an NP complete problem. Thus we must iterate it multiple times to get the optimal solution.